BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D. C. 20024

B70 03103

SUBJECT: Status Report on the Automated Task Scheduler (ATS) System - Case 610

DATE: March 31, 1970

FROM: A. B. Baker

ABSTRACT

The Automated Task Scheduler (ATS) System is a computer program being designed to produce timelines of in-flight activities for manned space missions at a level of detail normally found in a flight plan. Recent progress in the development of the ATS is described.

Program development has proceeded to the point where prototype schedules can be generated. In addition, a task data library has been written which contains preliminary descriptions of most of the Skylab in-flight activities, structured in the ATS data format. Finally, the capability to graphically display the crew, equipment, and consumables timelines has been implemented.

A representative schedule approximating the SL-1/2 mission is shown to demonstrate the system's present capabilities. Continuing development of priority structures, the scheduling algorithm, and timeline analysis capabilities is planned. Detailed program documentation is currently in preparation.

(NASA-CR-112619) STATUS REPORT ON THE AUTOMATED TASK SCHEDULER /ATS/ SYSTEM (Bellcomm, Inc.) 29 p

N79-73083

Unclas
00/54 11774

FF No. 602
CR-109443
(NASA CR OR TMX OR AD NUMBER) (CATEGORY)
[redacted]

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D.C. 20024

B70 03103

SUBJECT: Status Report on the Automated Task Scheduler (ATS) System - Case 610

DATE: March 31, 1970

FROM: A. B. Baker

MEMORANDUM FOR FILE

I. Introduction

The purpose of this memorandum is to summarize the status of the Automated Task Scheduler (ATS) System, a computer program which produces a nominal mission timeline (or schedule) for manned space missions at the level of detail normally found in a flight plan. The following discussion will be limited to a brief review of the system's construction, operational capabilities, and current status. Complete program documentation will be issued subsequently.

The incentive to develop an automated scheduler stems from the significant increase in the scope of future manned space missions compared to those in previous programs. Flight scheduling for the relatively short duration missions in the Mercury, Gemini, and Apollo programs was performed manually. However, missions in the Skylab Program are to last for one to two months and missions of even longer duration are being planned in the post-Skylab period. The increased duration of these missions will significantly increase the complexity of scheduling. It is this increase which provides the motivation for seeking to automate as much of the scheduling process as possible in order to: (1) reduce the burden of tedious manual scheduling, and (2) decrease the time required to construct detailed time-lines.

At the beginning of the effort, a review of the state-of-the-art found that a number of automated schedulers had already been built (Reference 1). However, all of the ones investigated had deficiencies, either in their basic algorithm or in their construction, which limited their utility; it was decided that a new scheduler could significantly improve on those already in existence. The review provided several concepts that have been used in the development of ATS. These include:

1. That the model be organized into three distinct functional areas -- Input and Data Preparation, Scheduler, and Processor.

2. That the model be sufficiently modular so that each function and major sub-function is as isolated as possible from the rest of the model. This structure facilitates evaluation of different computational techniques in each area.

3. That data libraries or "data banks" be established which would contain the large amounts of input data required by a scheduler. The banks, stored on magnetic tape or FASTRAND files, would simplify the input card decks for each computer run.

4. That spacecraft ephemeris data be generated independently of the scheduler.

5. That provision be made for analysis of timelines generated by the scheduler.

II. Alternative Scheduling Algorithms

The primary object of any automated scheduler is to assemble a given set of tasks into a self-consistent timeline within the structure defined by mission constraints, subsystem capabilities and inter-task constraints. There are basically two methods for achieving this objective: "sequential" timelining or timelining by "window-filling". Both are illustrated in Figure 1.

In the sequential approach, timelines are constructed by scheduling activities in chronological order; i.e., in a way analogous to adding successive links to a chain. The selection of the individual activity - or link - to be scheduled at a particular time is accomplished by choosing the highest priority activity (from the group of activities which have not yet been scheduled) whose requirements can be met at that point in the timeline. The task having been scheduled, mission time is incremented by an amount equal to the time required for the task, and the process is repeated. In the window-filling approach, each task is scheduled in descending order of priority by searching over the entire mission duration until a time is found (usually the first opportunity) when the task requirements are compatible with conditions already existing in the timeline.

There are two basic methods used to assign priorities to different tasks: static and dynamic. The word "static" is used to describe a method in which task priorities are assigned only once, before the scheduling process begins. The tasks are then scheduled anywhere over the mission duration, usually by the window-filling technique. In this way one would be assured that any time which was unavailable had already been committed to a task of higher priority.

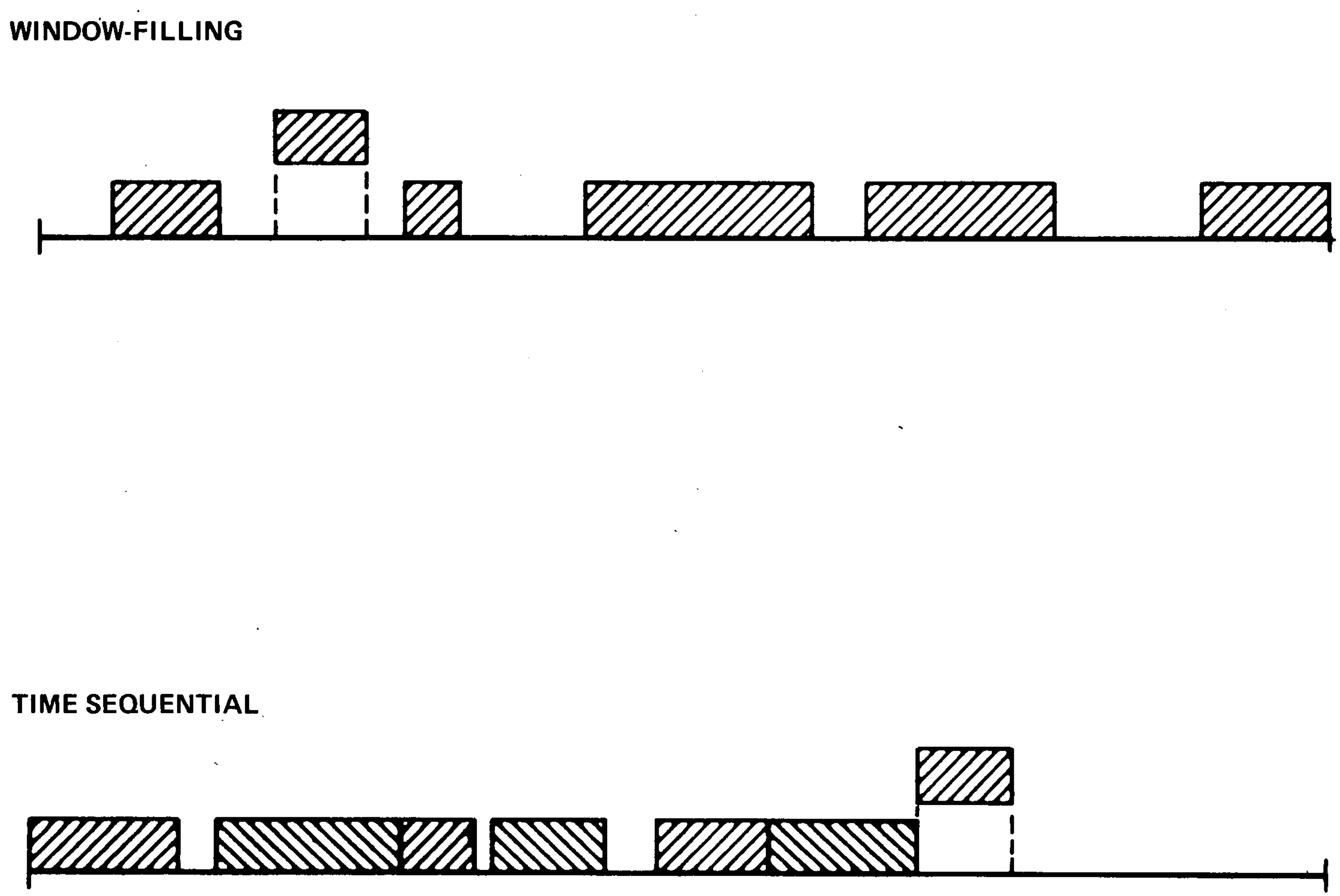


FIGURE 1 - ALTERNATIVE SCHEDULING CONCEPTS

The "dynamic" method is usually associated with sequential timelining. The dynamic method requires that the task priorities be recalculated at the beginning of each uncommitted time interval. The tasks are then reviewed in descending order of priority until one is found whose requirements are compatible with the current capabilities of the spacecraft and crew. The task is scheduled, mission time is advanced by the time required for the task, and the process is repeated.

III. The ATS Scheduling Algorithm

The window-filling technique was used in the construction of the ATS, primarily because it provides more flexibility than the sequential technique. Each task is defined by a group of requirements and performance constraints. However, since the ATS uses mission time as the basic independent variable, all requirements and constraints are translated into constraints on the time of performance. An acceptable interval for the performance of a particular task is therefore one during which all of the task requirements and performance constraints can be met.

Each task consists, in part, of a set of resource requirements. Each resource requirement exists for a definite interval relative to an arbitrary reference time called the "start time" of the task. Figure 2 illustrates a general set of requirements that would be specified in detail in the input data for each task. Requirement A might specify the services of a crewman; B a second crewman; C a level of electrical power, etc. As many as needed may be specified and the requirement intervals may have any desired position with respect to the task start-time. (Requirements prior to the start-time are perfectly acceptable.)

The ATS finds acceptable task start-time windows by sequentially overlaying the start-time windows determined by individual task requirements. The process is illustrated in Figure 3. When only the first requirement, Requirement A, is considered, the task may be initiated at any time within one of the three start-time windows: $A_1 - A_1'$, $A_2 - A_2'$, $A_3 - A_3'$. A range of acceptable start-times for the task based on consideration of Requirement B is then determined for the window $A_1 - A_1'$. In the case shown, the start-time window based only on Requirement B exceeds the limits established by the boundary points $A_1 - A_1'$. Therefore the boundary points $A_1 - A_1'$ now represent a permissible start-time window based upon consideration of both Requirements A and B. The boundary points of this

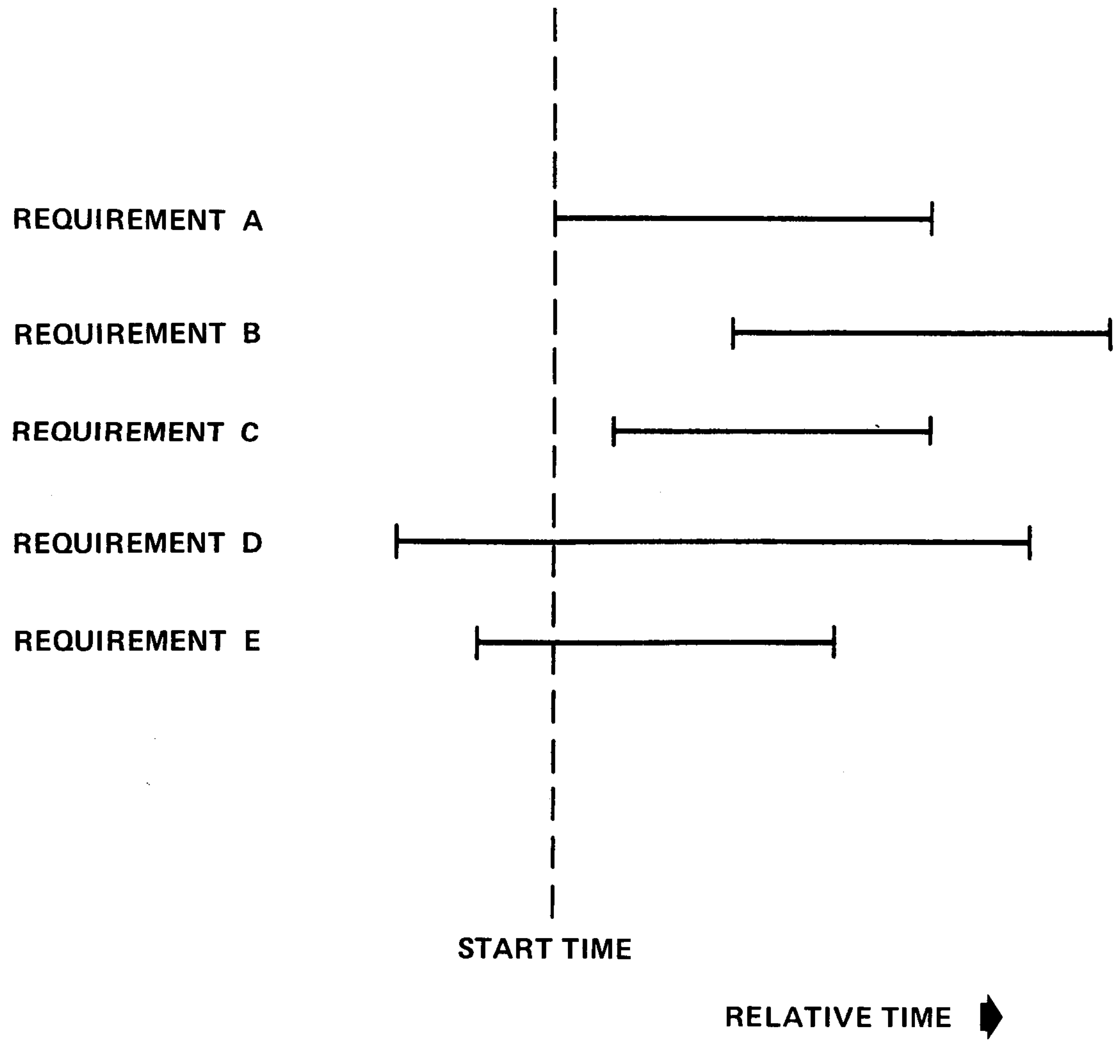


FIGURE 2 - TASK REQUIREMENT TIME DIAGRAM

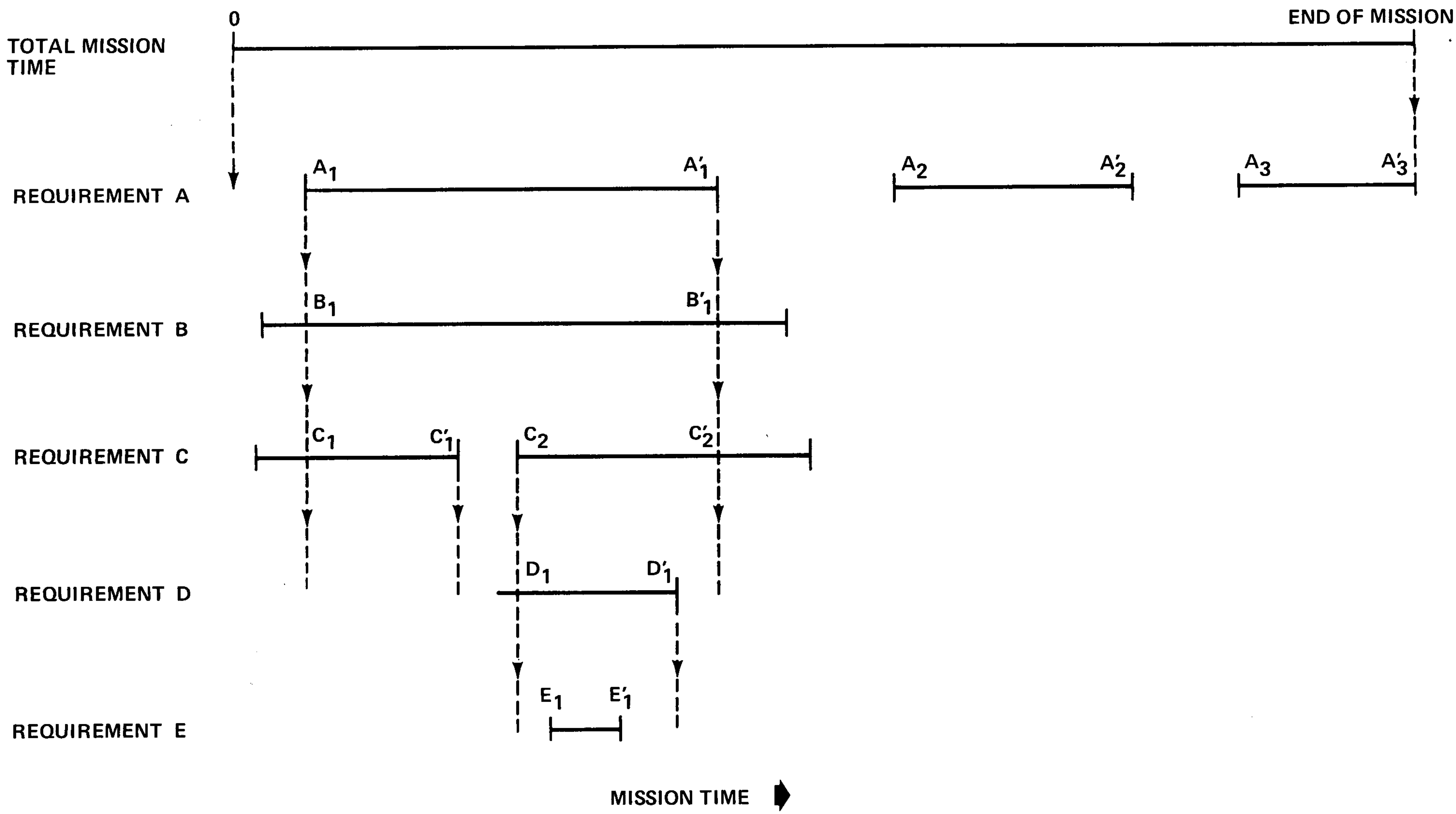


FIGURE 3 - DETERMINATION OF THE FIRST START-TIME WINDOW FOR A TASK HAVING REQUIREMENTS A, B, C, D, AND E

window are relabeled B_1 and B_1'. When Requirement C is considered over the range $B_1 - B_1'$, two separate start-time windows emerge, $C_1 - C_1'$ and $C_2 - C_2'$. The process is repeated again by considering Requirement D over the range of the window $C_1 - C_1'$. When no acceptable start times are found, the window $C_2 - C_2'$ is considered. When the last requirement, Requirement E, is considered over the range $D_1 - D_1'$, the window $E_1 - E_1'$ is obtained which represents the first acceptable start-time window for the task.

After finding the acceptable task start-time windows, the program schedules start-times for as many repetitions of the task as are required, by selecting points from within the start-time windows. If start-times can be found for the minimum number of performances required, the task is scheduled. Within the ATS, "scheduling" means updating the timelines of the required resources (crewman, equipment, consumables, etc.) to reflect the performance of the task. The major output of the scheduler is then a time history or timeline for each resource indicating where, in time, the resource is committed.

Note that each task is considered for scheduling only once during the generation of the schedule. At that time a decision is made whether to schedule the task at all (i.e., if acceptable start-time windows can be found) and if so which of the scheduling opportunities to utilize. These decisions are based only upon the commitments already made and do not consider the effect of the selections on future scheduling alternatives. No satisfactory method for implementing this capability (except, of course, exhaustive enumeration of alternatives) is yet apparent. Hence, for simplicity, the initial version of the ATS has been programmed to select the earliest possible start-time for the performance of a task. Other alternatives can, however, be easily evaluated and will be explored in future versions of the system.

IV. The ATS System

The overall flow diagram for the ATS System is shown in Figure 4. Structurally, the system follows the guidelines discussed in Section I. The system is organized into three structural areas:

1. The Input Section which loads input data from four different sources.

2. The Scheduler Section which combines the window-finding and scheduler functions described in Section III.

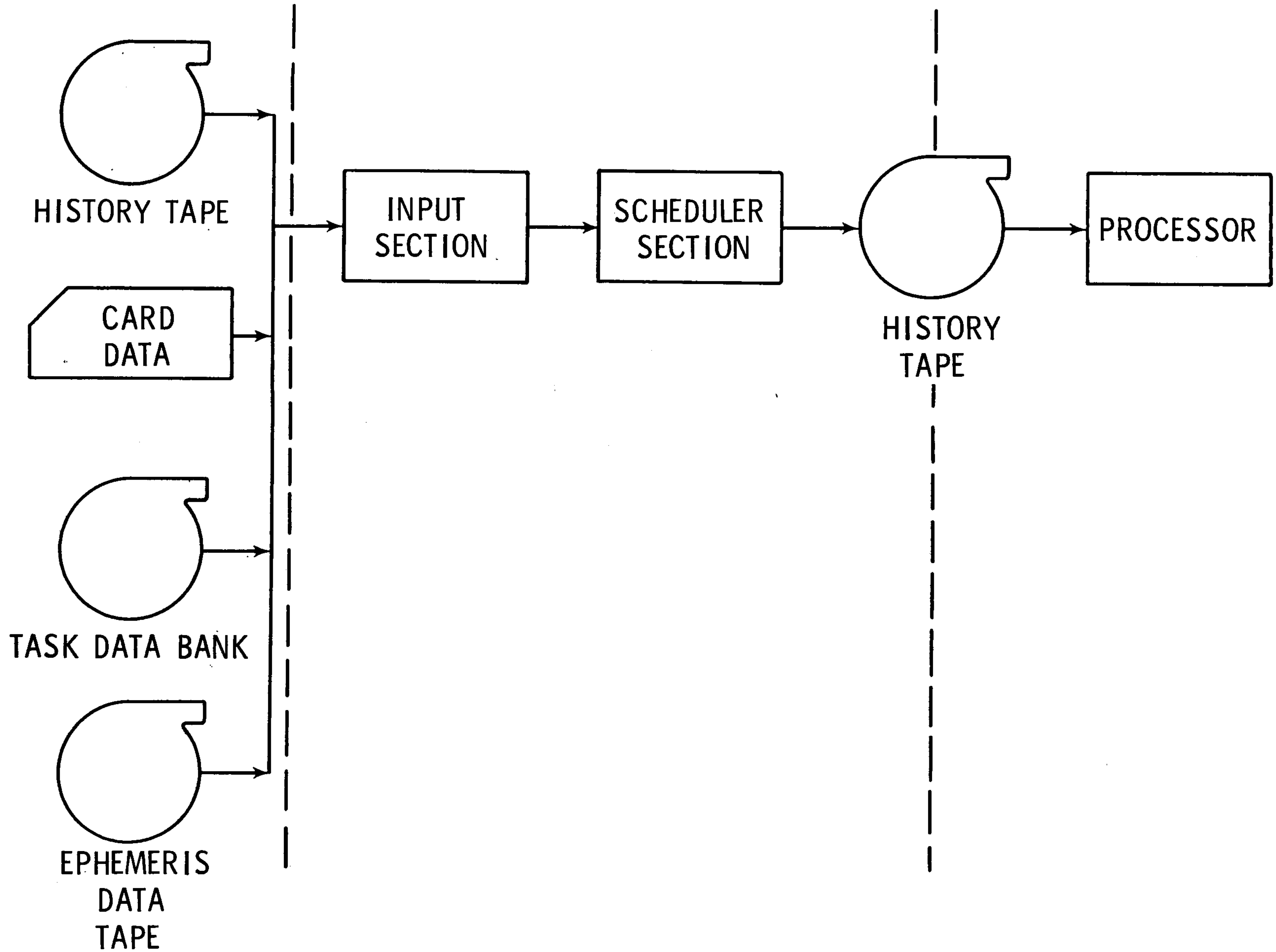


FIGURE 4 - ATS SYSTEM FLOW DIAGRAM

3. The Processor Section which has the capability to list and/or graphically display any of the timeline data generated and also to conduct the post-timeline analyses desired by the user.

The input and scheduling functions are combined in the basic scheduler. The primary output of the scheduler is a time history or timeline of the commitments for each resource and a list of start-times for each task. This output is printed out at the completion of the scheduler and also, on option, at regular intervals during the scheduling process. Also on option, the output will be written onto an output tape (labeled "History Tape" in Figure 4). The History Tape has two primary uses: to serve as the basic input to the Processor Section and to serve as an input to the scheduler itself on a subsequent run. The second use is particularly important. It enables the user to modify an existing schedule and hence eliminates the need to generate a completely new schedule on every run. This option can be used to conduct economical (in computer time and charges) investigations of variations in a basically desirable schedule. The History Tape is also used as the primary input to the Data Processor, thus permitting a complete or a partial schedule to be analyzed and displayed.

The four categories of input data accepted by the ATS are listed in Table 1. The Task Data Bank contains the descriptions of all of the tasks which have been accepted for the Skylab Program. These include personal tasks (e.g., sleep, breakfast, dinner, etc.), system housekeeping,tasks, and scientific experiments. When generating a schedule, the user specifies via input cards which of the tasks in the standard library are to be included in the schedule.

Four types of data are input via cards: mission characteristics, system constraints, program control cards, and task data modifications. The first three types are input in the first section of the data deck via a NAMELIST option. The second section permits the user to edit the tasks in the permanent library and/or to add new tasks for the particular run analogous to the way program elements from program files are edited temporarily for a particular computer run. To provide the capability of manipulating the task descriptions, a generalized card format, used for specifying task requirements and performance constraints, was developed for the ATS. It can be used for all of the functions mentioned above, i.e., to add, edit, or delete entire tasks or specific task requirements. The same card structure is used to create and maintain the Task Data Bank as well as to modify the task descriptions for a particular run. Details of the card structure and the associated editing

TABLE I

SCHEDULER INPUT DATA

DATA CATEGORY	CONTENT
HISTORY TAPE	PARTIAL SCHEDULE FROM A PRIOR RUN
DATA BANK TAPE	EXPERIMENT DESCRIPTIONS FROM THE STANDARD LIBRARY
CARD	SYSTEM CONSTRAINTS MISSION CHARACTERISTICS PROGRAM CONTROL CARDS TASK DATA MODIFICATIONS
EPHEMERIS DATA TAPE	EPHEMERIS DERIVED DATA TABLES •SPACECRAFT SUNLIGHT •TRACKING STATION CONTACTS •PHOTOGRAPHIC TARGET CONTACTS

procedures will be presented as part of the program documentation. Finally, ephemeris information, generated by the BCMASP Earth-Orbit Simulator (Reference 2), is stored on a magnetic tape which in turn is used as an input to the ATS.

During the execution of the program, there are wide variations in the amount of data generated by many of the sub-functional areas (e.g., the number of task specifications, the number of start-time windows generated, the number of entries in each resource timeline history). Using only FORTRAN IV, it would be necessary to allot to each data array the maximum size expected for the duration of the run, despite the fact that this maximum is seldom needed. The result is an unnecessarily large and inefficient allocation of computer core space. To achieve efficient use of core space and hence reduce the overall size of the ATS program, a dynamic system of core allocation was achieved by using a FORTRAN-imbedded list-processing language named SAC-1 (Reference 3). Details of this implementation will also be presented as part of the program documentation.

V. Current Status

The basic scheduler (Input and Scheduler Sections) with all of the capabilities described in Section III has recently become operational and is presently capable of generating schedules. This portion of the ATS consists of 48 functions and subroutines totaling approximately 32,700 words. In addition, preliminary versions of the primary requirements and performance constraints for virtually all of the tasks normally scheduled in Skylab Program Flight Plans have been translated into the data structure required by the ATS and have been placed in a Permanent Data Bank. Finally, a portion of the Data Processor was constructed which is capable of displaying resource timeline histories in graphic form. The latter uses the AUPLOT system (Reference 4) as an interface to the SC-4020 plotter. The current version of the Processor, though limited in capability, is sufficiently flexible to permit the user to select the particular resources for which timelines are to be plotted, the total number to be plotted on a single axis, and the scale to which these timelines are to be plotted.

To demonstrate the present capabilities of the ATS, the system was used to produce a representative schedule for the SL-1/2 mission. Task descriptions for the tasks listed in Table 2 were input to the ATS as scheduling candidates along with the mission characteristics listed in Table 3. As Table 2 indicates, all but three of the tasks were scheduled. Graphs of the resulting crew timelines, produced by the Data Processor to a scale of 2 days per frame, are shown in Figure 5. Note that timelines were produced for only 25 days, since it was

TABLE 2

SL-1/2 CANDIDATE TASKS

BREAK	Breakfast
DINNER	
*D008	Radiation in Spacecraft
EVADOD	EVA for DOD Experiments
EVAATM	EVA for ATM Experiments
LUNCH	
M092	In-flight Lower Body Negative Pressure
M093	In-flight Vectorcardiogram
M131	Human Vestibular Function
M171	Metabolic Activity
M172	Body Mass Measurement (included in Breakfast)
M172C	Calibration for Body Mass Measurement
**M508	Astronaut EVA Hardware Evaluation
M509	Astronaut Maneuvering Equipment
REST	
SETUP1	Activation Activities
SETUP2	Activation Activities
SHKP	System Housekeeping
SLEEP	
S019	Ultraviolet Stellar Astronomy
S020	X-Ray/UV Solar Photography
T003	In-flight Nephelometer
T020	Foot-Controlled Maneuvering Unit
T025	Coronagraph Contamination Measurements
†T027	Contamination Measurement

*Portion inside the S.A. Anomaly could not be scheduled

**Could not be scheduled

†Portion concerned with measurement of sky background illumination could not be scheduled.

TABLE 3

SL-1/2 MISSION CHARACTERISTICS

SL-1 Launch Time	14:58 hours (EST)
SL-1 Launch Date	July 16, 1972
Number of Crewmen	3
Mission Duration	28 days
Scheduling Restrictions	
Activation Activities	Day 1
Deactivation Activities	Days 26 and 27
Deorbit Activities	Day 28

assumed at least two full days would be required for deactivation and deorbit activities. In addition, activation activities were assumed for the first 30 hrs in orbit. (Activation activities are denoted by the tasks SETUP1 and SETUP2 which are performed during the first mission day.)

As stated above, the timelines shown in Figure 5 are presented to illustrate the capabilities of the ATS and should not be construed as a recommendation for an actual flight plan. The illustrated timelines reflect the present lack of a complete and accurate data base. For example, the schedule does not include all of the tasks approved for SL-1/2 nor does it reflect the latest changes to the requirements and constraints of the scientific experiment tasks. An effort is already under way (Reference 5) to provide a more complete and up-to-date data base which can be used to generate meaningful schedules.

VI. Future Effort

Future development of the ATS System will include:

1. The construction of a full Data Processing Section which will provide a choice of output graph formats as well as the capability for post-timeline analysis.

2. Improvements to the window-finder algorithm and to the Task Data Bank to improve their flexibility.

3. The evaluation of different methods of implementing a "look ahead" capability and incorporating these results into the calculation of dynamic priorities.

Allan Baker

1025-ABB-li A. B. Baker

Attachments

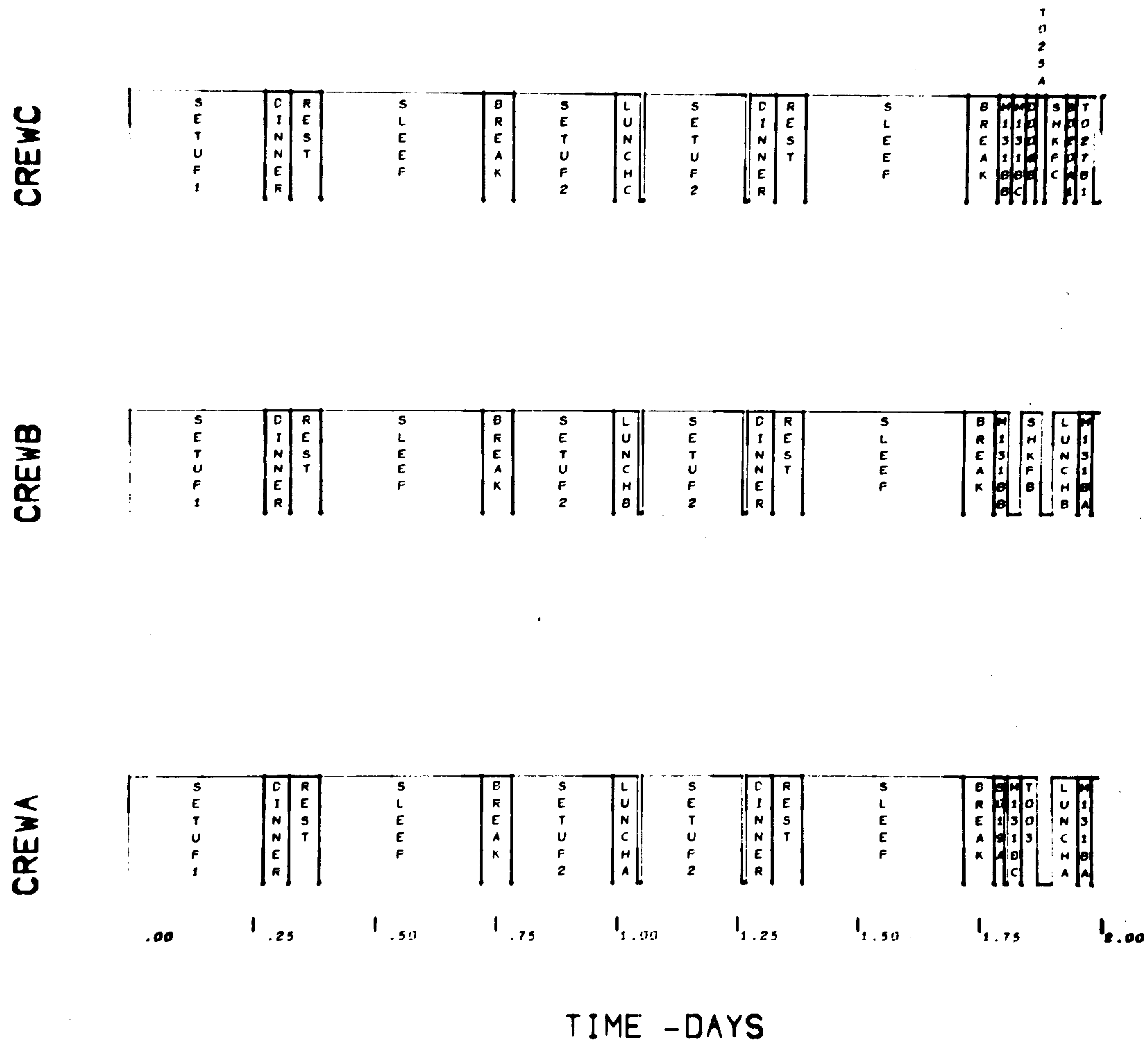


FIGURE 5 - SL 1/2 CREW TIMELINES

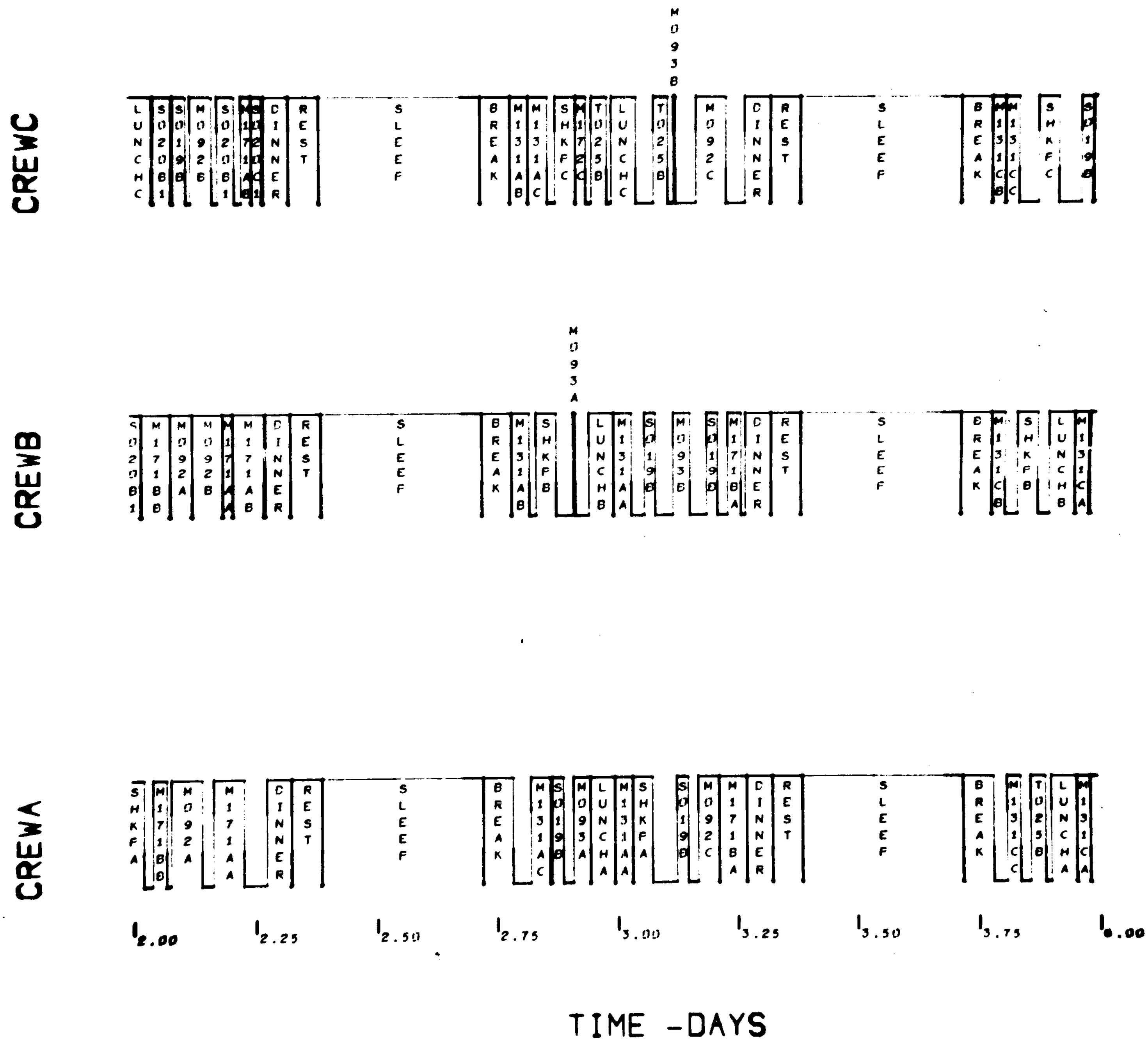
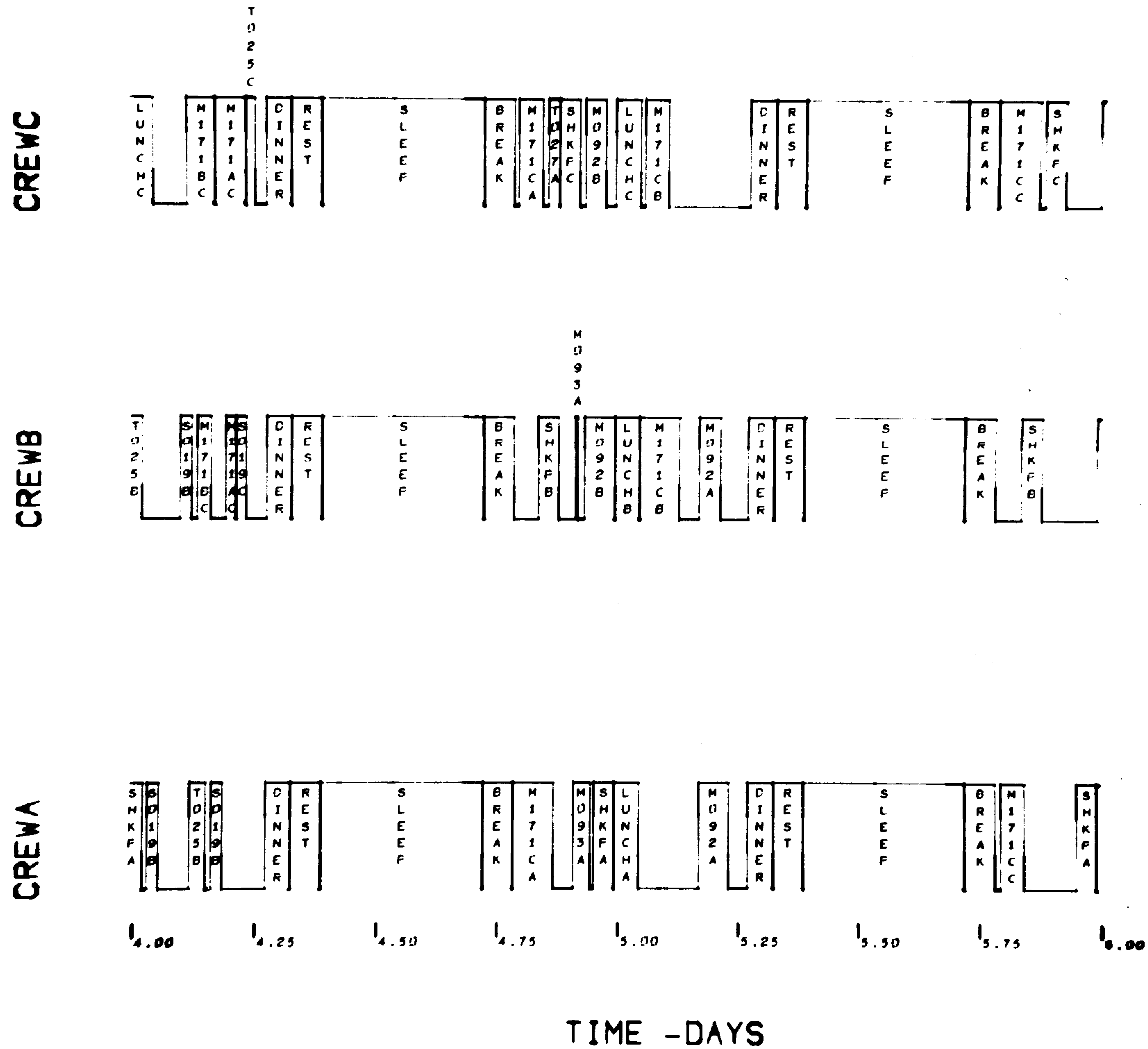
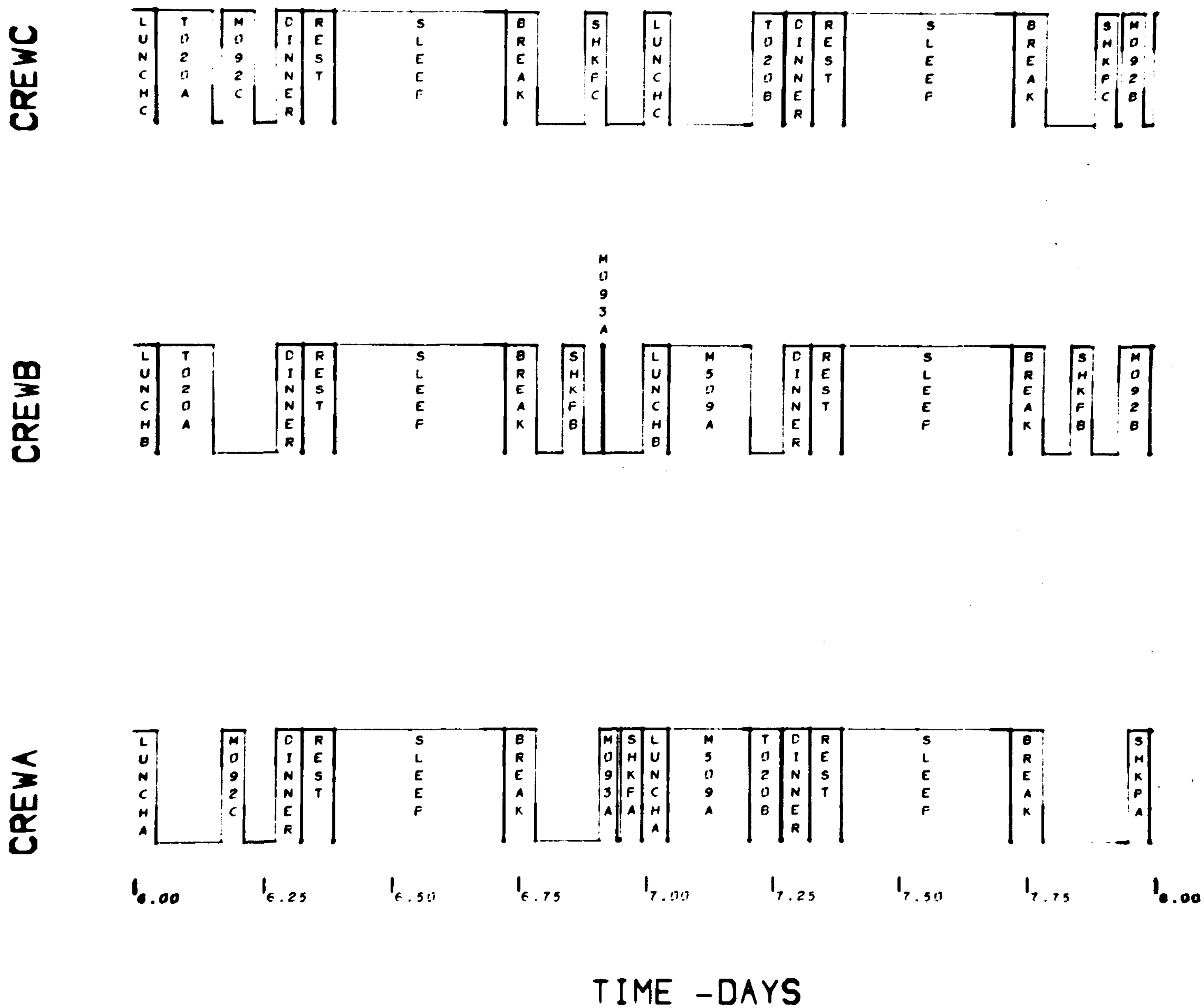

CREWC
CREWB
CREWA
2.00
2.25
2.50
2.75
3.00
3.25
3.50
3.75
4.00
TIME -DAYS

FIGURE 5
PAGE 3 OF 13




CREWC
LUNCHC
T020A
M092C
DINNER
REST
SLEEP
BREAK
SHKFC
LUNCHC
T020B
DINNER
REST
SLEEP
BREAK
SHKFC
M092B
CREWB
LUNCHB
T020A
DINNER
REST
SLEEP
BREAK
SHKFB
M093A
LUNCHB
M509A
DINNER
REST
SLEEP
BREAK
SHKFB
M092B
CREWA
LUNCHA
M092C
DINNER
REST
SLEEP
BREAK
M093A
SHKFA
LUNCHA
M509A
T020B
DINNER
REST
SLEEP
BREAK
SHKPA
6.00
6.25
6.50
6.75
7.00
7.25
7.50
7.75
8.00
TIME -DAYS

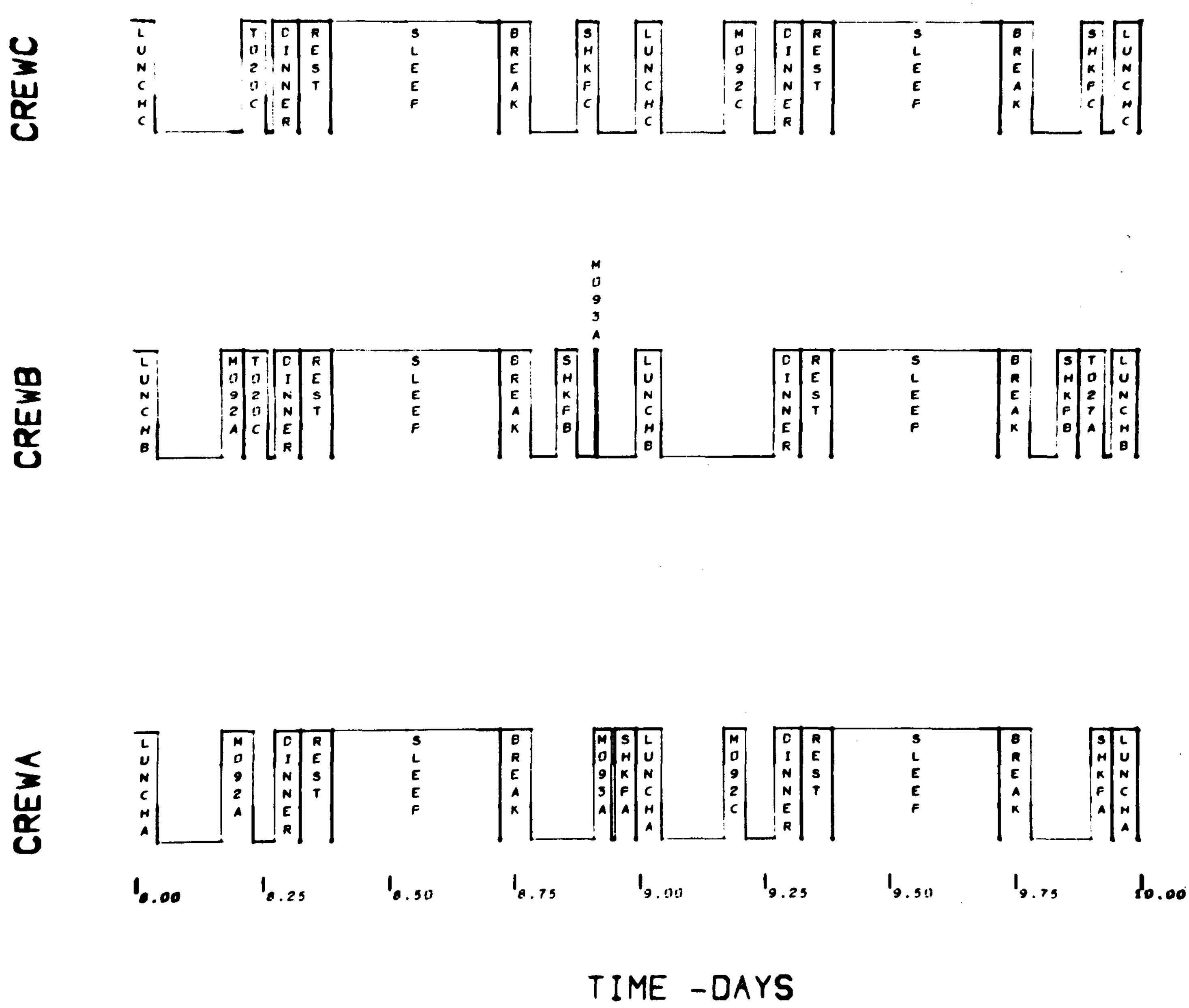

CREWC
LUNCHC
TO20C
DINNER
REST
SLEEP
BREAK
SHKFC
LUNCHC
MO92C
DINNER
REST
SLEEP
BREAK
SHKFC
LUNCHC
CREWB
LUNCHB
MO92A
TO20C
DINNER
REST
SLEEP
BREAK
SHKFB
MO93A
LUNCHB
DINNER
REST
SLEEP
BREAK
SHKFB
TO27A
LUNCHB
CREWA
LUNCHA
MO92A
DINNER
REST
SLEEP
BREAK
MO93A
SHKFA
LUNCHA
MO92C
DINNER
REST
SLEEP
BREAK
SHKFA
LUNCHA
8.00
8.25
8.50
8.75
9.00
9.25
9.50
9.75
10.00
TIME -DAYS

FIGURE 5
PAGE 6 OF 13

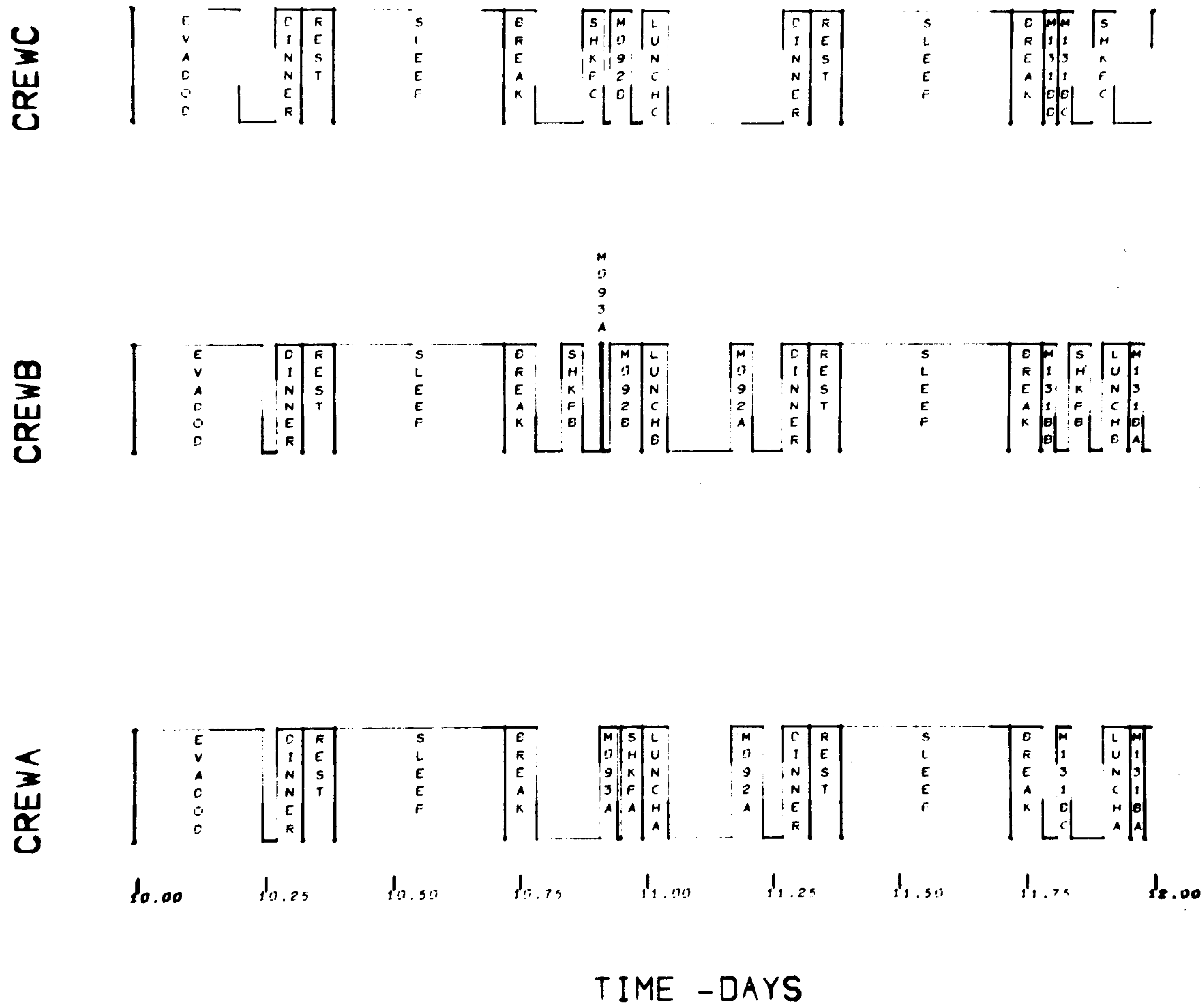


FIGURE 5
PAGE 7 OF 13

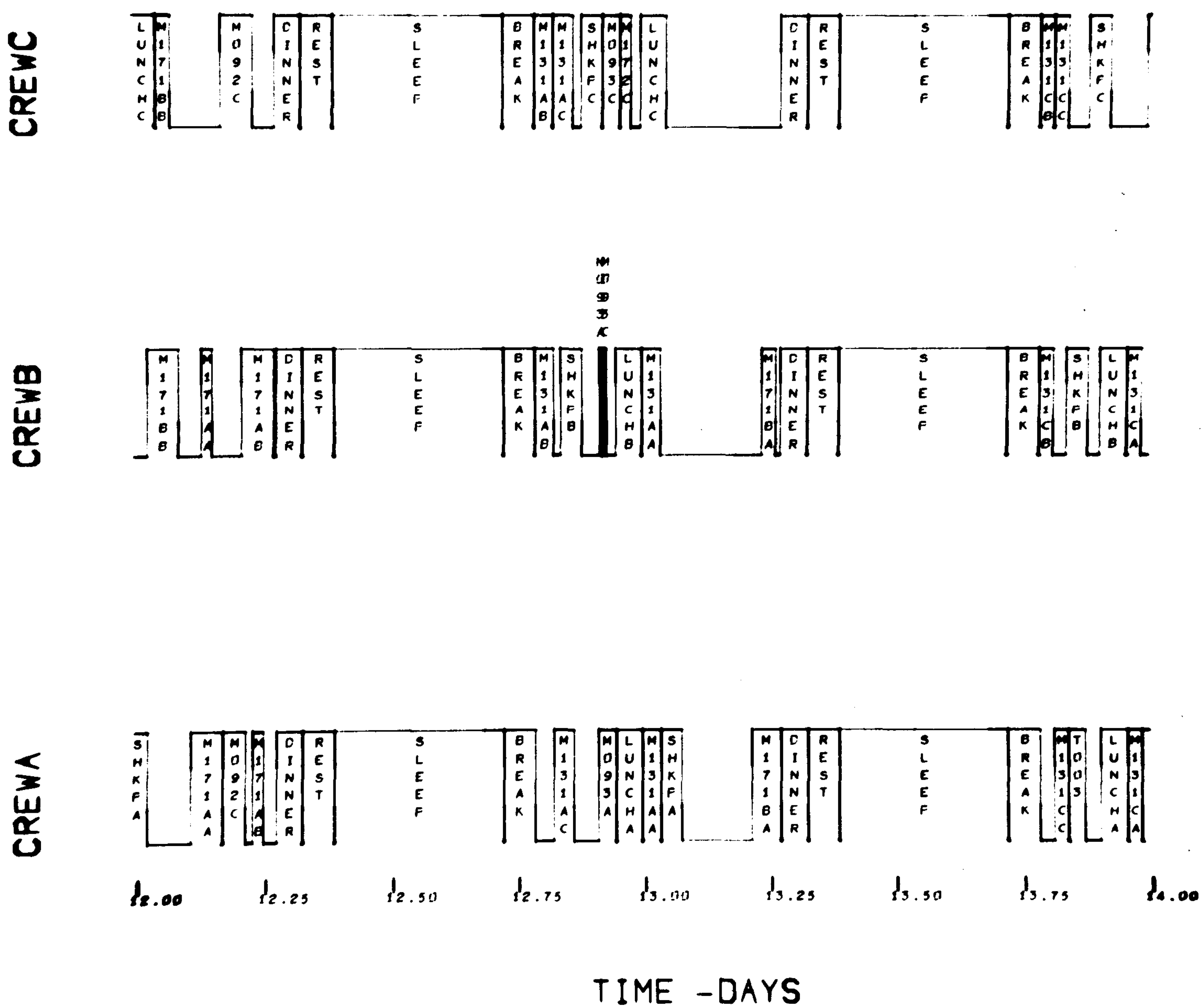


FIGURE 5
PAGE 8 OF 13

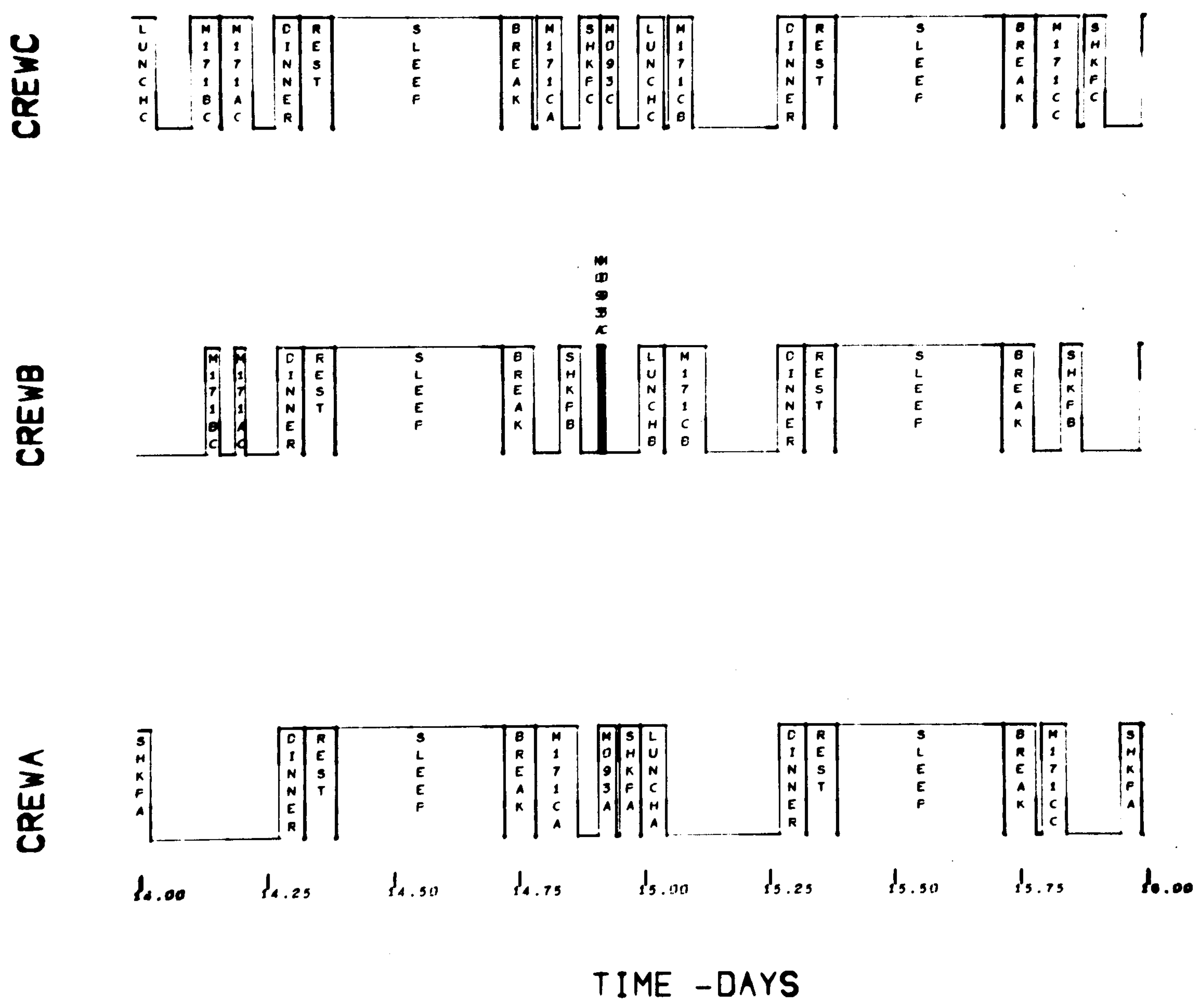


FIGURE 5
PAGE 9 OF 13

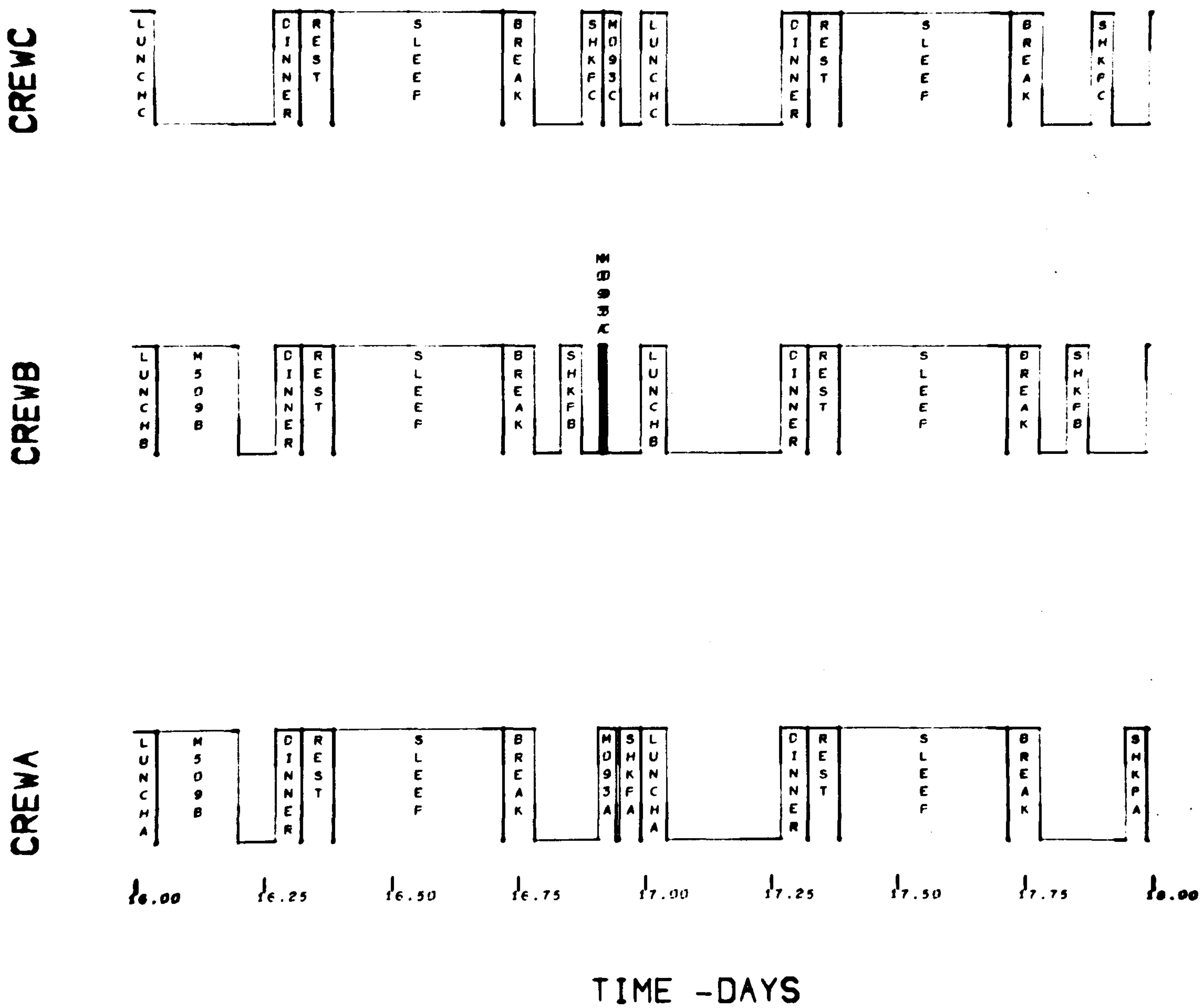


FIGURE 5
PAGE 9 OF 13

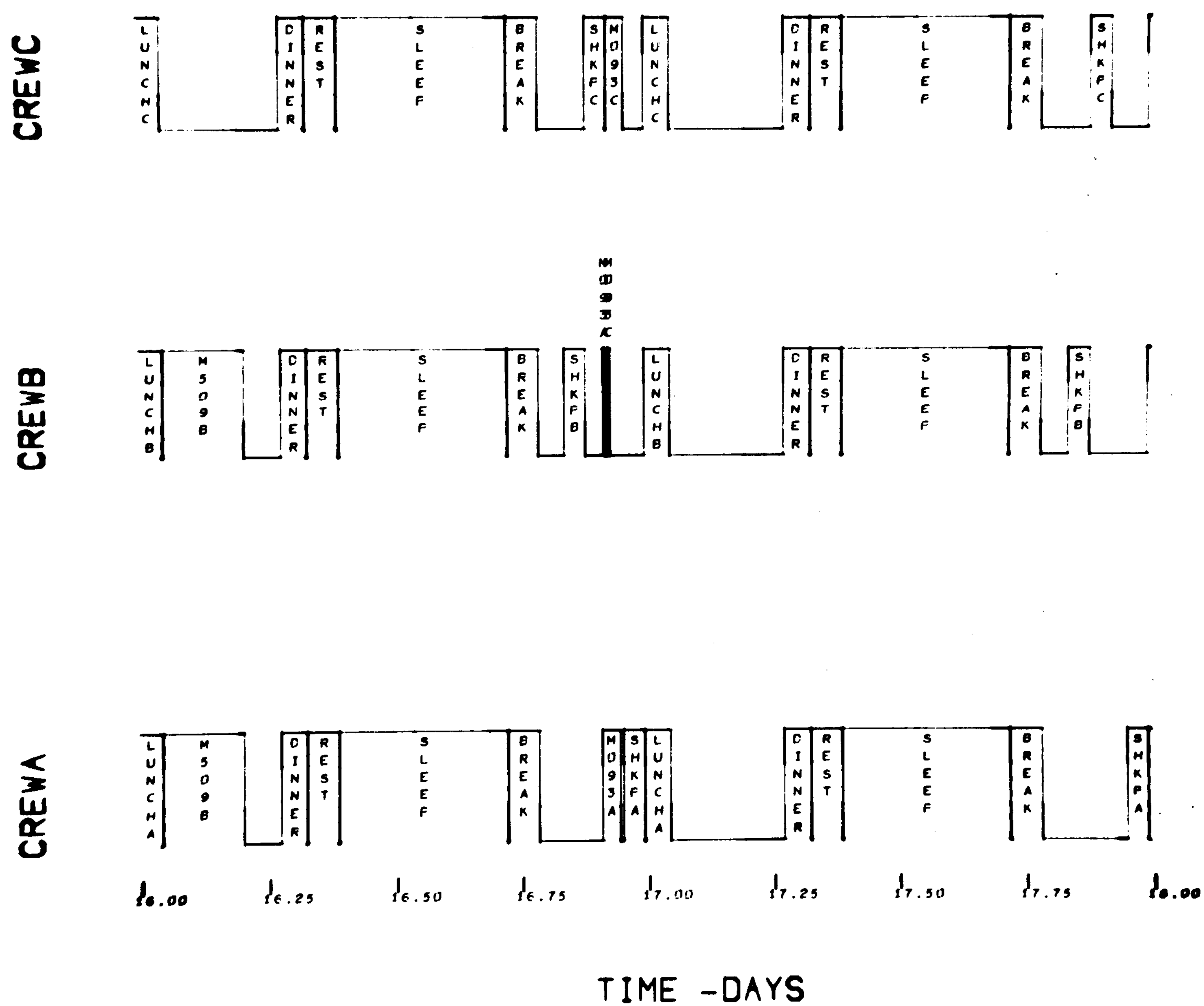
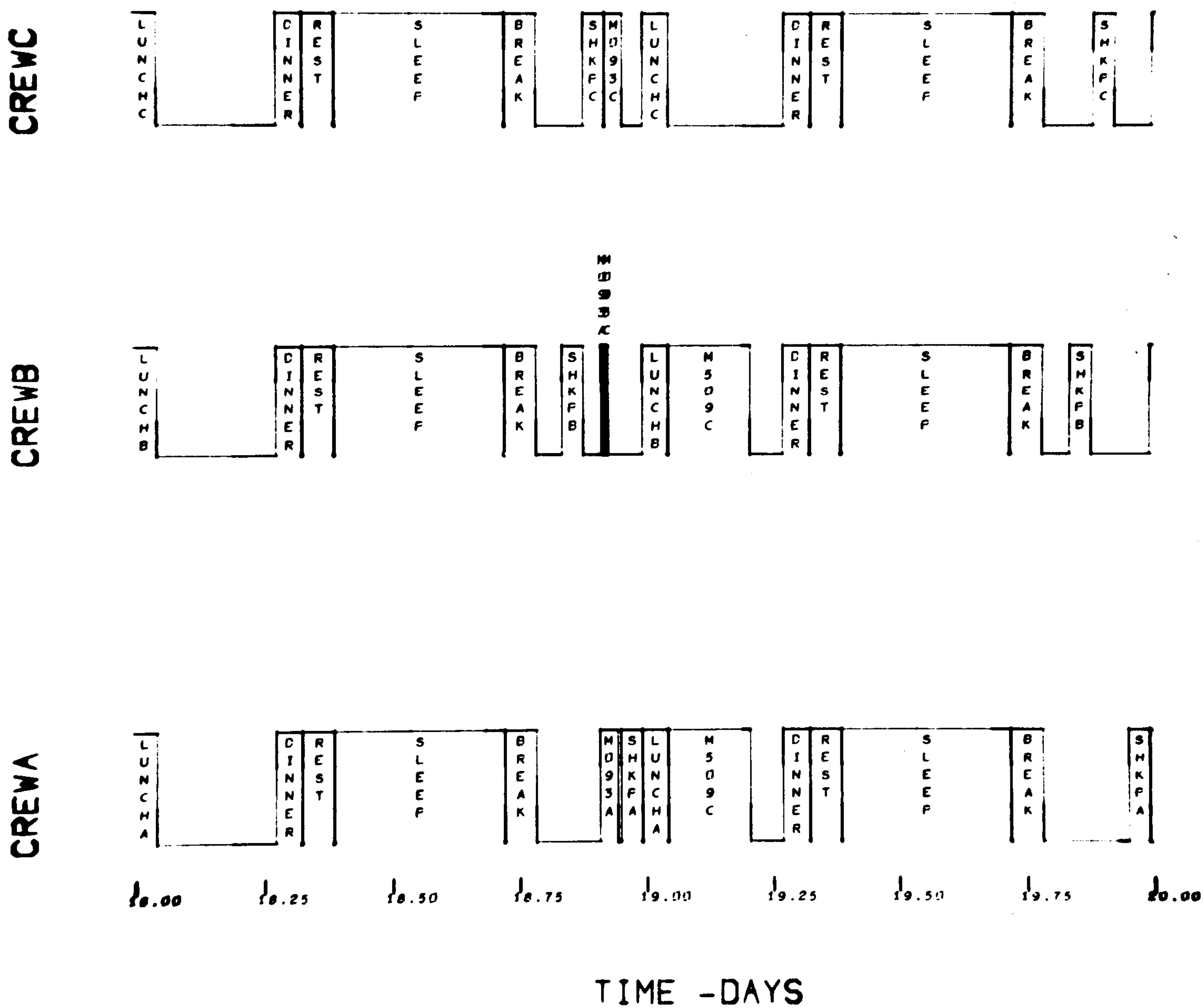


FIGURE 5
PAGE 10 OF 13



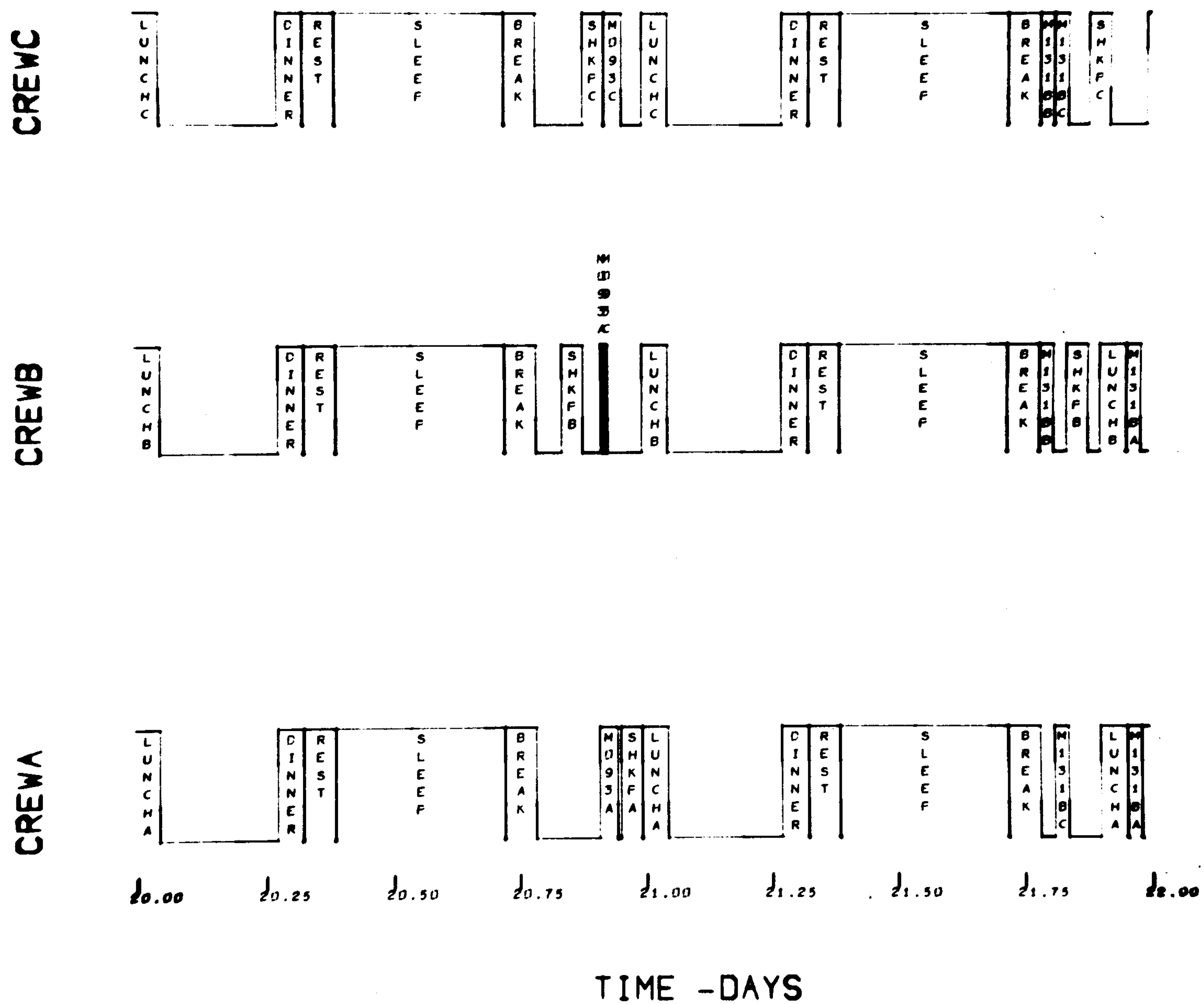

CREWC
CREWB
CREWA
LUNCHC
DINNER
REST
SLEEP
BREAK
SHKFC
LUNCHB
SHKFB
LUNCHA
SHKFA
TIME -DAYS

FIGURE 5
PAGE 12 OF 13

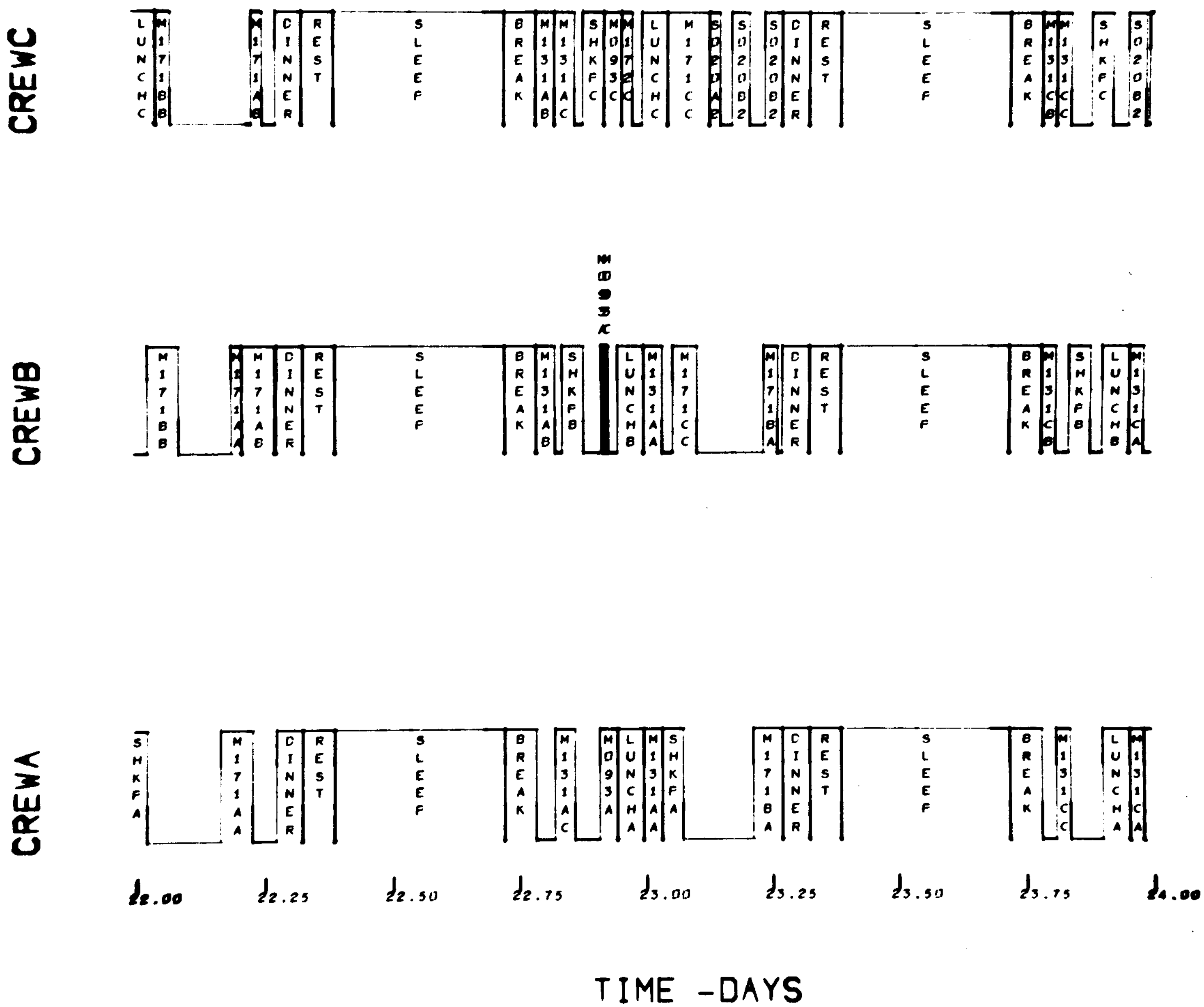
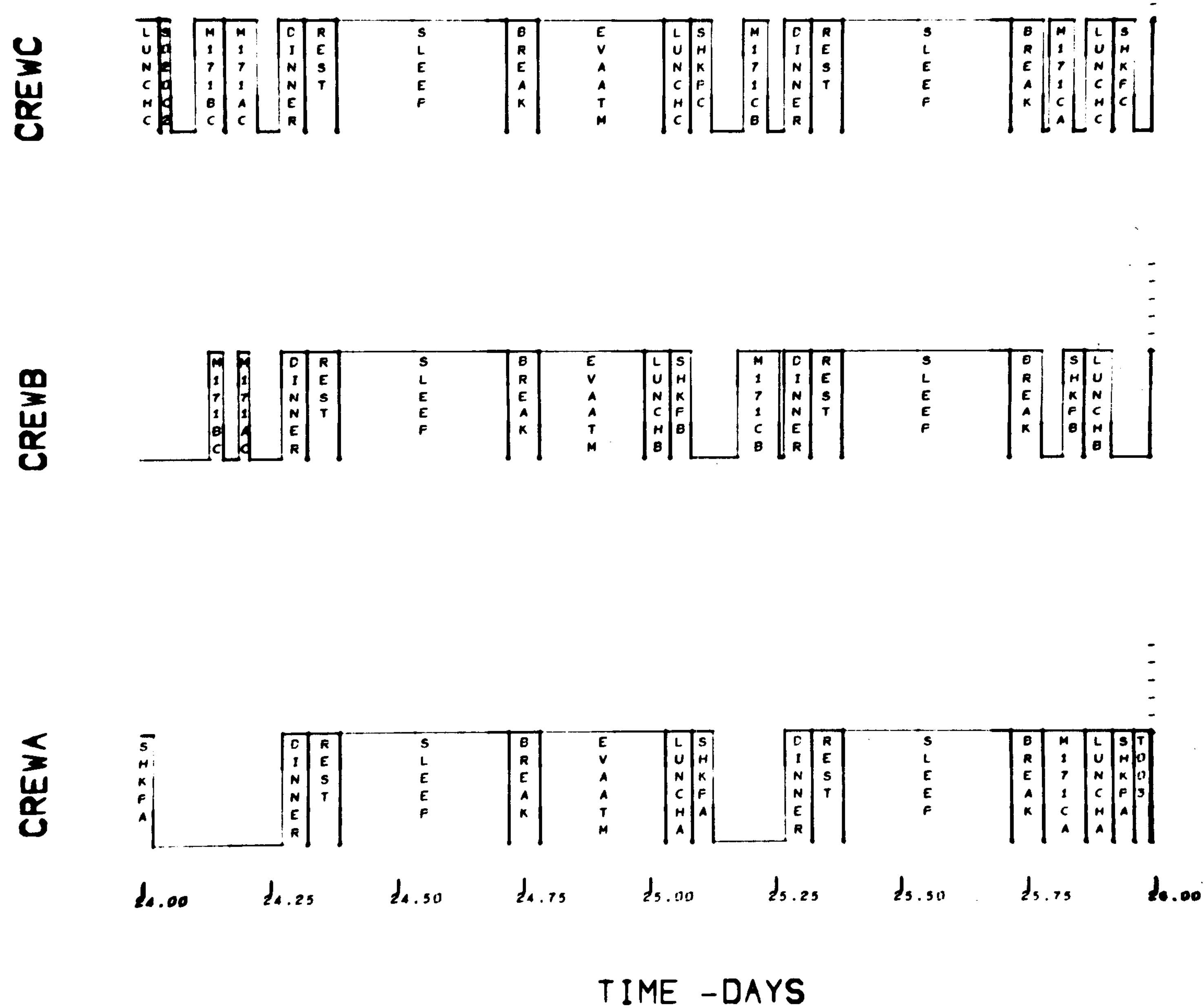


FIGURE 5


CREWC
LUNCHC
M171BC
M171AC
DINNER
REST
SLEEP
BREAK
EVAATM
LUNCHC
SHKFC
M171CB
DINNER
REST
SLEEP
BREAK
M171CA
LUNCHC
SHKFC
CREWB
M171BC
M171AC
DINNER
REST
SLEEP
BREAK
EVAATM
LUNCHB
SHKFB
M171CB
DINNER
REST
SLEEP
BREAK
SHKFB
LUNCHB
CREWA
SHKFA
DINNER
REST
SLEEP
BREAK
EVAATM
LUNCHA
SHKFA
DINNER
REST
SLEEP
BREAK
M171CA
LUNCHA
SHKFA
TOD3
24.00
24.25
24.50
24.75
25.00
25.25
25.50
25.75
26.00
TIME -DAYS

REFERENCES

1. A. B. Baker
   "A Survey of Automated Scheduling Models"
   Bellcomm B69 04020 April 7, 1969

2. A. B. Baker
   "Revisions to the BCMASP Earth-Orbit Simulator Program"
   Bellcomm B69 11020 November 10, 1969

3. M. P. Odle
   "The SAC-1 System for List Processing and Formula Manipulation"
   Bellcomm B70 04001 April 1, 1970

4. R. F. Jessup
   "The AUPLOT Graphical Data Processing System"
   Bellcomm B69 10117 October 27, 1969

5. B. H. Crane
   "Status Report on Experiment Descriptions for the Automated Task Scheduler (ATS) Program"
   Bellcomm In Preparation

BELLCOMM, INC.

Subject: Status Report on the Automated Task Scheduler (ATS) System - Case 610

From: A. B. Baker

Distribution List

NASA Headquarters

H. Cohen/MLR
J. H. Disher/MLD
W. B. Evans/MLO
L. K. Fero/MLV
J. P. Field, Jr./MLP
T. E. Hanes/MLA
V. N. Huff/MTE
T. A. Keegan/MA-2
M. Savage/MLT
W. C. Schneider/ML
D. N. Turner/MTE

Bellcomm, Inc.

A. P. Boysen, Jr.
J. O. Cappellari, Jr.
D. R. Hagner
W. G. Heffron
H. A. Helm
J. J. Hibbert
B. T. Howard
R. F. Jessup
P. F. Long
J. Z. Menard
L. D. Nelson
J. M. Nervik
M. P. Odle
I. M. Ross
P. F. Sennewald
R. V. Sperry
J. W. Timko
R. L. Wagner
M. P. Wilson
Division 101 Supervision
All Members Division 102
Department 1024 File
Central Files
Library